Knowledge Representation.

Propositional logic.

Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square

Knowledge-based agent

- **Knowledge base (KB):**
  - A set of sentences that describe facts about the world in some formal (representational) language
  - **Domain specific**
- **Inference engine:**
  - A set of procedures that use the representational language to infer new facts from known ones or answer a variety of KB queries. Inferences typically require search.
  - **Domain independent**
Example: MYCIN

- MYCIN: an expert system for diagnosis of bacterial infections
- **Knowledge base** represents
  - Facts about a specific patient case
  - Rules describing relations between entities in the bacterial infection domain

| If | 1. The stain of the organism is gram-positive, and |
|    | 2. The morphology of the organism is coccus, and |
|    | 3. The growth conformation of the organism is chains |
| Then | the identity of the organism is streptococcus |

- **Inference engine:**
  - manipulates the facts and known relations to answer diagnostic queries (consistent with findings and rules)

Knowledge representation

- The objective of knowledge representation is to express the knowledge about the world in a computer-tractable form

- Key aspects of knowledge representation languages:
  - **Syntax:** describes how sentences are formed in the language
  - **Semantics:** describes the meaning of sentences, what is it the sentence refers to in the real world
  - **Computational aspect:** describes how sentences and objects are manipulated in concordance with semantical conventions

Many KB systems rely on some variant of logic
Logic

A formal language for expressing knowledge and for making logical inferences.

Logic is defined by:

- **A set of sentences**
  - A sentence is constructed from a set of primitives according to syntax rules.

- **A set of interpretations**
  - An interpretation gives a semantic to primitives. It associates primitives with values.

- **The valuation (meaning) function** $V$
  - Assigns a value (typically the truth value) to a given sentence under some interpretation.

\[ V : \text{sentence} \times \text{interpretation} \rightarrow \{ \text{True}, \text{False} \} \]

Propositional logic

- **The simplest logic**

- **Definition**: A proposition is a statement that is either true or false.

- **Examples**: Pitt is located in the Oakland section of Pittsburgh.
  - (T)
Propositional logic

• The simplest logic

• **Definition:**
  – A proposition is a statement that is either true or false.

• Examples:
  – Pitt is located in the Oakland section of Pittsburgh.
    • (T)
  – 5 + 2 = 8.
    • ?
Propositional logic

• The simplest logic

• **Definition:**
  – A **proposition** is a statement that is either true or false.

• Examples:
  – Pitt is located in the Oakland section of Pittsburgh.
    • (T)
  – $5 + 2 = 8$.
    • (F)
  – It is raining today.
    • (either T or F)

• Examples (cont.):
  – How are you?
    • ?
Propositional logic

• Examples (cont.):
  – How are you?
    • a question is not a proposition
  – $x + 5 = 3$
    • ?

• Since $x$ is not specified, neither true nor false
• $2$ is a prime number.
  • ?
Propositional logic

• Examples (cont.):
  – How are you?
    • a question is not a proposition
  – $x + 5 = 3$
    • since $x$ is not specified, neither true nor false
  – 2 is a prime number.
    • (T)
  – She is very talented.
    • ?

• Examples (cont.):
  – How are you?
    • a question is not a proposition
  – $x + 5 = 3$
    • since $x$ is not specified, neither true nor false
  – 2 is a prime number.
    • (T)
  – She is very talented.
    • since she is not specified, neither true nor false
  – There are other life forms on other planets in the universe.
    • ?
Propositional logic

• Examples (cont.):
  – How are you?
    • a question is not a proposition
  – $x + 5 = 3$
    • since $x$ is not specified, neither true nor false
  – 2 is a prime number.
    • (T)
  – She is very talented.
    • since she is not specified, neither true nor false
  – There are other life forms on other planets in the universe.
    • either T or F

Propositional logic. Syntax

• Formally propositional logic $P$:
  – Is defined by Syntax+interpretation+semantics of $P$

Syntax:

• Symbols (alphabet) in $P$:
  – Constants: True, False
  – Propositional symbols
    Examples:
    • $P$
      • Pitt is located in the Oakland section of Pittsburgh.,
    • It rains outside, etc.
  – A set of connectives:
    $\neg, \land, \lor, \Rightarrow, \Leftrightarrow$
Propositional logic. Syntax

Sentences in the propositional logic:

• Atomic sentences:
  – Constructed from constants and propositional symbols
  – True, False are (atomic) sentences
  – \( P \cdot Q \) or Light in the room is on, It rains outside are (atomic) sentences

• Composite sentences:
  – Constructed from valid sentences via connectives
  – If \( A, B \) are sentences then
    \( \neg A \) (\( A \land B \)) (\( A \lor B \)) (\( A \Rightarrow B \)) (\( A \Leftrightarrow B \))
    or (\( A \lor B \)) \( \land \) (\( A \lor \neg B \))
    are sentences

Propositional logic. Semantics.

The semantic gives the meaning to sentences.

the semantics in the propositional logic is defined by:

1. **Interpretation of propositional symbols and constants**
   – Semantics of atomic sentences

2. **Through the meaning of connectives**
   – Meaning (semantics) of composite sentences
Semantic: propositional symbols

A propositional symbol

- a statement about the world that is either true or false

Examples:
  - Pitt is located in the Oakland section of Pittsburgh
  - It rains outside
  - Light in the room is on

- An interpretation maps symbols to one of the two values: True (T), or False (F), depending on whether the symbol is satisfied in the world

  I: Light in the room is on -> True, It rains outside -> False

  I': Light in the room is on -> False, It rains outside -> False

Semantic: propositional symbols

The meaning (value) of the propositional symbol for a specific interpretation is given by its interpretation

  I: Light in the room is on -> True, It rains outside -> False

  \[ V(\text{Light in the room is on, } I) = \text{True} \]

  \[ V(\text{It rains outside, } I) = \text{False} \]

  I': Light in the room is on -> False, It rains outside -> False

  \[ V(\text{Light in the room is on, } I') = \text{False} \]
Semantics: constants

- **The meaning (truth) of constants:**
  - True and False constants are always (under any interpretation) assigned the corresponding True, False value

\[
\begin{align*}
V(True, I) &= True \\
V(False, I) &= False
\end{align*}
\]

For any interpretation \( I \)

---

Semantics: composite sentences.

- **The meaning (truth value) of complex propositional sentences.**
  - Determined using the standard rules of logic:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>(-P)</th>
<th>(P \land Q)</th>
<th>(P \lor Q)</th>
<th>(P \Rightarrow Q)</th>
<th>(P \Leftrightarrow Q)</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
</tbody>
</table>
Translation

Translation of English sentences to propositional logic:
(1) identify atomic sentences that are propositions
(2) Use logical connectives to translate more complex composite sentences that consist of many atomic sentences

Assume the following sentence:
• It is not sunny this afternoon and it is colder than yesterday.

Atomic sentences:
• p = It is sunny this afternoon
• q = it is colder than yesterday

Translation: \( \neg p \land q \)

Translation

Assume the following sentences:
• It is not sunny this afternoon and it is colder than yesterday.
• We will go swimming only if it is sunny.
• If we do not go swimming then we will take a canoe trip.
• If we take a canoe trip, then we will be home by sunset.

Denote:
• p = It is sunny this afternoon
• q = it is colder than yesterday
• r = We will go swimming
• s= we will take a canoe trip
• t= We will be home by sunset
Translation

Assume the following sentences:
• It is not sunny this afternoon and it is colder than yesterday. \( \neg p \land q \)
• We will go swimming only if it is sunny.
• If we do not go swimming then we will take a canoe trip.
• If we take a canoe trip, then we will be home by sunset.

Denote:
• \( p \) = It is sunny this afternoon
• \( q \) = it is colder than yesterday
• \( r \) = We will go swimming
• \( s \) = we will take a canoe trip
• \( t \) = We will be home by sunset
Translation

Assume the following sentences:
• It is not sunny this afternoon and it is colder than yesterday. \( \neg p \land q \)
• We will go swimming only if it is sunny. \( r \rightarrow p \)
• If we do not go swimming then we will take a canoe trip. \( \neg r \rightarrow s \)
• If we take a canoe trip, then we will be home by sunset. \( s \rightarrow t \)

Denote:
• \( p \) = It is sunny this afternoon
• \( q \) = it is colder than yesterday
• \( r \) = We will go swimming
• \( s \) = we will take a canoe trip
• \( t \) = We will be home by sunset
Contradiction and Tautology

Some composite sentences may always (under any interpretation) evaluate to a single truth value:

- **Contradiction** (always *False*)
  \[ P \land \neg P \]

- **Tautology** (always *True*)
  \[ P \lor \neg P \]

\[
\begin{align*}
\neg(P \lor Q) & \iff (\neg P \land \neg Q) \\
\neg(P \land Q) & \iff (\neg P \lor \neg Q)
\end{align*}
\]

DeMorgan’s Laws

Model, validity and satisfiability

- A **model (in logic)**: An interpretation is a model for a set of sentences if it assigns true to each sentence in the set.
- A sentence is **satisfiable** if it has a model;
  - There is at least one interpretation under which the sentence can evaluate to True.
- A sentence is **valid** if it is *True* in all interpretations
  - i.e., if its negation is **not satisfiable** (leads to contradiction)

<table>
<thead>
<tr>
<th>( P )</th>
<th>( Q )</th>
<th>( P \lor Q )</th>
<th>( (P \lor Q) \land \neg Q )</th>
<th>( (P \lor Q) \land \neg Q \Rightarrow P )</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>True</td>
</tr>
</tbody>
</table>
Model, validity and satisfiability

- A **model (in logic)**: An interpretation is a model for a set of sentences if it assigns true to each sentence in the set.
- A sentence is **satisfiable** if it has a model;
  - There is at least one interpretation under which the sentence can evaluate to True.
- A sentence is **valid** if it is *True* in all interpretations
  - i.e., if its negation is not satisfiable (leads to contradiction)

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>P ∨ Q</th>
<th>(P ∨ Q) ∧ ¬Q</th>
<th>((P ∨ Q) ∧ ¬Q) ⇒ P</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>True</td>
</tr>
</tbody>
</table>

Model, validity and satisfiability

- A **model (in logic)**: An interpretation is a model for a set of sentences if it assigns true to each sentence in the set.
- A sentence is **satisfiable** if it has a model;
  - There is at least one interpretation under which the sentence can evaluate to True.
- A sentence is **valid** if it is *True* in all interpretations
  - i.e., if its negation is not satisfiable (leads to contradiction)

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>P ∨ Q</th>
<th>(P ∨ Q) ∧ ¬Q</th>
<th>((P ∨ Q) ∧ ¬Q) ⇒ P</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>True</td>
</tr>
</tbody>
</table>
Model, validity and satisfiability

• A **model (in logic)**: An interpretation is a model for a set of sentences if it assigns true to each sentence in the set.
• A sentence is **satisfiable** if it has a model;
  – There is at least one interpretation under which the sentence can evaluate to True.
• A sentence is **valid** if it is **True** in all interpretations
  – i.e., if its negation is **not satisfiable** (leads to contradiction)

<table>
<thead>
<tr>
<th>Satisfiable sentence</th>
<th>Valid sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P \lor Q$</td>
<td>$((P \lor Q) \land \neg Q) \Rightarrow P$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>$P \lor Q$</th>
<th>$(P \lor Q) \land \neg Q$</th>
<th>$((P \lor Q) \land \neg Q) \Rightarrow P$</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td>True</td>
</tr>
</tbody>
</table>

Entailment

• **Entailment** reflects the relation of one fact in the world following from the others

- $KB \models \alpha$
- Knowledge base $KB$ entails sentence $\alpha$ if and only if $\alpha$ is true in all worlds where $KB$ is true
Sound and complete inference.

Inference is a process by which conclusions are reached.
- We want to implement the inference process on a computer !!

Assume an inference procedure \( i \) that
- derives a sentence \( \alpha \) from the KB: \( KB \models_i \alpha \)

Properties of the inference procedure in terms of entailment
- **Soundness:** An inference procedure is sound  
  If \( KB \models_i \alpha \) then it is true that \( KB \models \alpha \)
- **Completeness:** An inference procedure is complete  
  If \( KB \models \alpha \) then it is true that \( KB \models_i \alpha \)

Logical inference problem

**Logical inference problem:**
- **Given:**
  - a knowledge base KB (a set of sentences) and
  - a sentence \( \alpha \) (called a theorem),
- **Does a KB semantically entail \( \alpha \)? \( KB \models \alpha \)?
In other words: In all interpretations in which sentences in the KB are true, is also \( \alpha \) true?

**Question:** Is there a procedure (program) that can decide this problem in a finite number of steps?
**Answer:** Yes. Logical inference problem for the propositional logic is **decidable**.
Solving logical inference problem

In the following:

**How to design the procedure that answers:**

\[ KB \models \alpha \ ? \]

**Three approaches:**
- Truth-table approach
- Inference rules
- Conversion to the inverse SAT problem
  - Resolution-refutation

---

**Truth-table approach**

**Problem:** \[ KB \models \alpha \ ? \]

- We need to check all possible interpretations for which the KB is true (models of KB) whether \( \alpha \) is true for each of them

**Truth table:**
- enumerates truth values of sentences for all possible interpretations (assignments of True/False values to propositional symbols)

**Example:**

<table>
<thead>
<tr>
<th>( P )</th>
<th>( Q )</th>
<th>( P \lor Q )</th>
<th>( P \iff Q )</th>
<th>( (P \lor \neg Q) \land Q )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>True</strong></td>
<td><strong>True</strong></td>
<td><strong>True</strong></td>
<td><strong>True</strong></td>
<td><strong>True</strong></td>
</tr>
<tr>
<td><strong>True</strong></td>
<td><strong>False</strong></td>
<td><strong>True</strong></td>
<td><strong>False</strong></td>
<td><strong>False</strong></td>
</tr>
<tr>
<td><strong>False</strong></td>
<td><strong>True</strong></td>
<td><strong>True</strong></td>
<td><strong>False</strong></td>
<td><strong>False</strong></td>
</tr>
<tr>
<td><strong>False</strong></td>
<td><strong>False</strong></td>
<td><strong>False</strong></td>
<td><strong>True</strong></td>
<td><strong>False</strong></td>
</tr>
</tbody>
</table>
Truth-table approach

**Problem:** \( KB \models \alpha \) ?

- We need to check all possible interpretations for which the KB is true (models of KB) whether \( \alpha \) is true for each of them

**Truth table:**

- enumerates truth values of sentences for all possible interpretations (assignments of True/False to propositional symbols)

**Example:**

<table>
<thead>
<tr>
<th>( P )</th>
<th>( Q )</th>
<th>( P \lor Q )</th>
<th>( P \Leftrightarrow Q )</th>
<th>( (P \lor \neg Q) \land Q )</th>
<th>( \alpha )</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
</tbody>
</table>

 Truth-table approach

**Problem:** \( KB \models \alpha \) ?

- We need to check all possible interpretations for which the KB is true (models of KB) whether \( \alpha \) is true for each of them

**Truth table:**

- enumerates truth values of sentences for all possible interpretations (assignments of True/False to propositional symbols)

**Example:**

<table>
<thead>
<tr>
<th>( P )</th>
<th>( Q )</th>
<th>( P \lor Q )</th>
<th>( P \Leftrightarrow Q )</th>
<th>( (P \lor \neg Q) \land Q )</th>
<th>( \alpha )</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
</tbody>
</table>

\( \checkmark \)
Truth-table approach

A two steps procedure:

1. Generate table for all possible interpretations
2. Check whether the sentence $\alpha$ evaluates to true whenever $KB$ evaluates to true

Example: $KB = (A \lor C) \land (B \lor \neg C)$ $\alpha = (A \lor B)$

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>$A \lor C$</th>
<th>$(B \lor \neg C)$</th>
<th>$KB$</th>
<th>$\alpha$</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
</tbody>
</table>
Truth-table approach

A two steps procedure:
1. Generate table for all possible interpretations
2. Check whether the sentence $\alpha$ evaluates to true whenever $KB$ evaluates to true

Example: $KB = (A \lor C) \land (B \lor \neg C)$ $\alpha = (A \lor B)$

<table>
<thead>
<tr>
<th>$A$</th>
<th>$B$</th>
<th>$C$</th>
<th>$A \lor C$</th>
<th>$(B \lor \neg C)$</th>
<th>$KB$</th>
<th>$\alpha$</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
</tbody>
</table>

KB entails $\alpha$

• The truth-table approach is sound and complete for the propositional logic!!